NEXT JS

CONTENTS

- Introduction to Next JS ,Moving from React to Next JS

- Creating Next JS App

- Navigation between pages

- Managing Assets,MetaData and Css

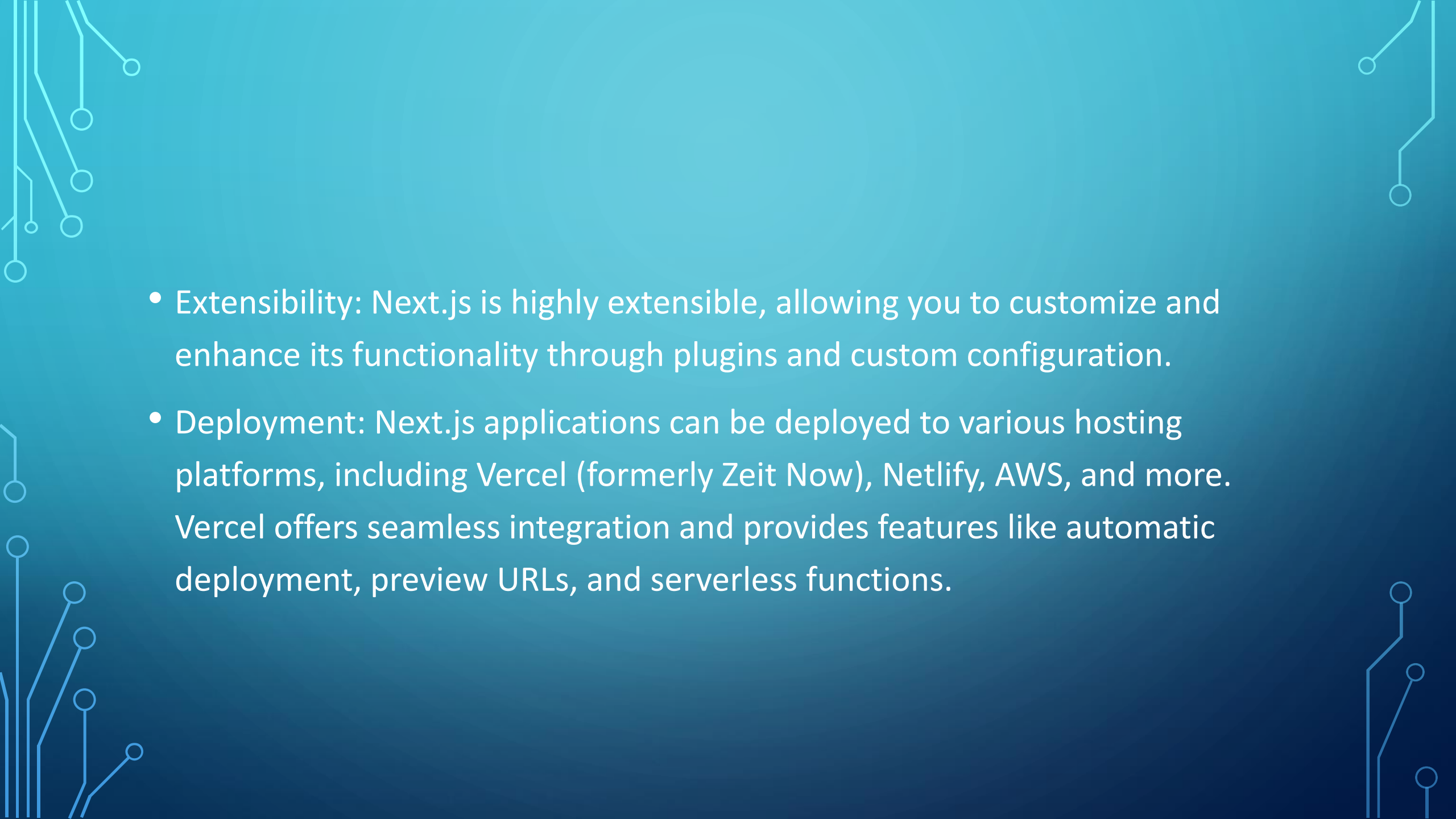- Pre rendering and Data Fetching

- Dynamic Routes

- Best Practices

INTRODUCTION TO NEXT JS

- Next.js is a popular open-source framework for building modern web applications using React. It is designed to provide a robust and efficient development experience for creating server-rendered, statically generated, and single-page applications.

SOME KEY FEATURES AND CONCEPTS OF NEXT.JS

- Server-Side Rendering (SSR): Next.js allows you to render React components on the server and send HTML to the client. This enables faster initial page loads and improves search engine optimization (SEO).

- Static Site Generation (SSG): Next.js can generate static HTML files at build time, which can be served directly from a content delivery network (CDN). This approach is ideal for content that doesn't change frequently, as it eliminates the need for server-side rendering on every request.

- Client-Side Rendering (CSR): Next.js supports client-side rendering, which allows for dynamic updates and interactivity on the client side after the initial page load.

- Automatic Code Splitting: Next.js automatically splits your JavaScript code into smaller chunks, enabling faster page loads by only loading the necessary code for each page.

- Routing: Next.js has a file-based routing system. You can create routes by simply creating pages inside a directory, without the need for additional configuration.

- CSS Support: Next.js supports various CSS methodologies, including CSS modules, CSS-in-JS libraries like styled-components and emotion, and global stylesheets.

- Extensibility: Next.js is highly extensible, allowing you to customize and enhance its functionality through plugins and custom configuration.

- Deployment: Next.js applications can be deployed to various hosting platforms, including Vercel (formerly Zeit Now), Netlify, AWS, and more. Vercel offers seamless integration and provides features like automatic deployment, preview URLs, and serverless functions.

- Next.js has gained popularity due to its developer-friendly features, performance optimizations, and flexibility in building modern web applications. It simplifies complex tasks and provides a solid foundation for creating fast, scalable, and SEO-friendly applications.

CREATING A NEXT JS APP

- To create next js app execute below command

- npx create-next-app@latest <app-name>

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\peter>d:

D:\>cd nextjs

D:\nextjs>npx create-next-app@latest myblogapp
√ Would you like to use TypeScript with this project? ... No / Yes
√ Would you like to use ESLint with this project? ... No / Yes
√ Would you like to use Tailwind CSS with this project? ... No / Yes
√ Would you like to use `src/` directory with this project? ... No / Yes
√ Use App Router (recommended)? ... No / Yes
√ Would you like to customize the default import alias? ... No / Yes
Creating a new Next.js app in D:\nextjs\myblogapp.

Using npm.

Initializing project with template: app



Installing dependencies:
- react
- react-dom
- next



added 20 packages, and audited 21 packages in 12s

4 packages are looking for funding
```

NEXT JS APP PROJECT STRUCTURE

- Inside application ,the most important folder is app folder that contains all the required files for application

- Global.css

- Layout.js

- Page.js

- Page.module.css

EXECUTE THE APPLICATION IN DEVELOPMENT MODE

- npm run dev

```
PS D:\nextjs\myblogapp> npm run dev

> myblogapp@0.1.0 dev
> next dev

- ready started server on 0.0.0.0:3000, url: http://localhost:3000
- event compiled client and server successfully in 5.2s (306 modules)
- wait compiling...
- event compiled client and server successfully in 2.2s (306 modules)
- wait compiling /page (client and server)...
- event compiled client and server successfully in 15.6s (592 modules)
```
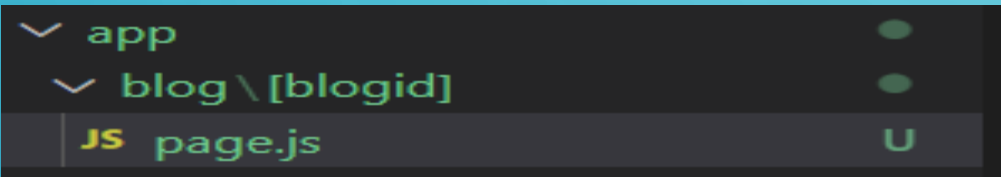
CREATING A COMPONENT IN NEXT JS

- A component in next js can be created in the app directory or inside a sub directory

NAVIGATION IN NEXT JS

- NEXT JS follows file based routing system in which you have to create a directory and place the component in a file named page.js/page.tsx

- Next JS also provides for dynamic routes ,for this a subfolder with name in square braces can be used to set the name of route parameter

```
∨ app
  ∨ blog\[blogid]
    JS page.js                        U
```

export default function Page({params})

{

  return <h1>Blog No {params.blogid}</h1>

}

LINK TAG

- A Link tag can be created to navigate to the component using Link component

CLIENT SIDE AND SERVER SIDE COMPONENT

- By default component in next js are server side components ,

- This does not allow for client side code to be executed directly

- To execute client side code we have to use clause

- "use client"

```
"use client"

export default function Page({params})
{
    return (<div>
        <h1>Blog No {params.blogid}</h1>
        <button onClick={()=>{
            alert('hello user')
        }}>Click</button>
    </div>)
}
```

DATA FETCHING IN COMPONENT

```
export default async function Page({params})

{

  const id=params.blogid

   const res =await fetch(`https://jsonplaceholder.typicode.com/posts/${id}`)

  const data=await res.json()

  console.log(data)

  return (<div>

     <h1>Blog No {data.id} {data.title}</h1>


   </div>)

}
```